**Information processing cycle:**

The computer system converts data into information by performing various activities:

* Receive data
* Process it
* Display processed data
* Store it

**Input:** accepts data from the user through the keyboard or storage unit.

**Processing**: Raw data from various sources is processed step by step according to instructions, by the CPU (Central processing unit) which carries out both arithmetic & logical operations.

**Output:** Sending processes data to output devices like printers. The data can also be stored directly on a storage disk without being displayed. Text, graphics, sound

**Storage:** The computer can store the processed data permanently in storage devices for future use. Hard disk, floppy, CD

**Block diagram of a Computer:**

Modern computer systems consist of both hardware & software based on Von Neumann architecture. Essential hardware required to perform operations:

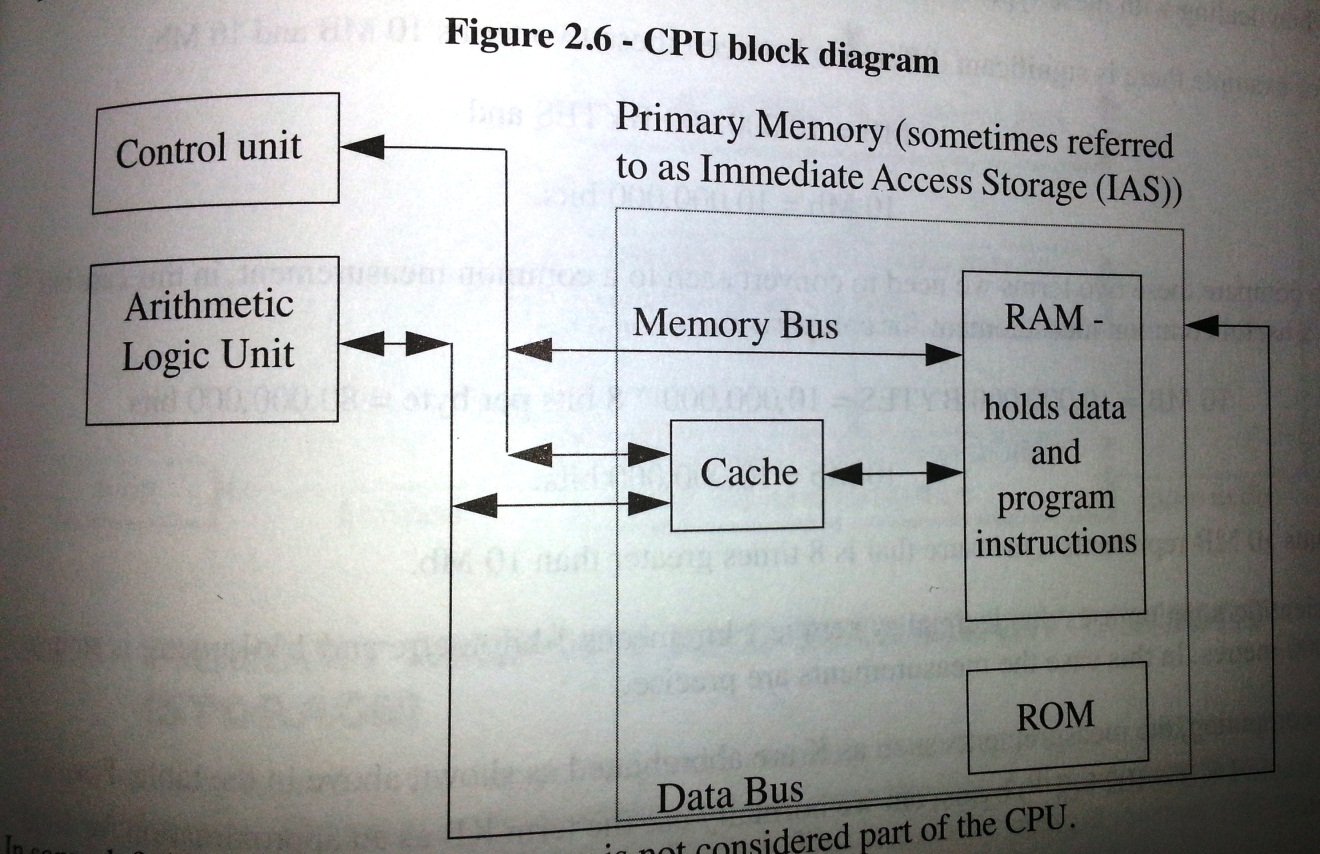
Input unit

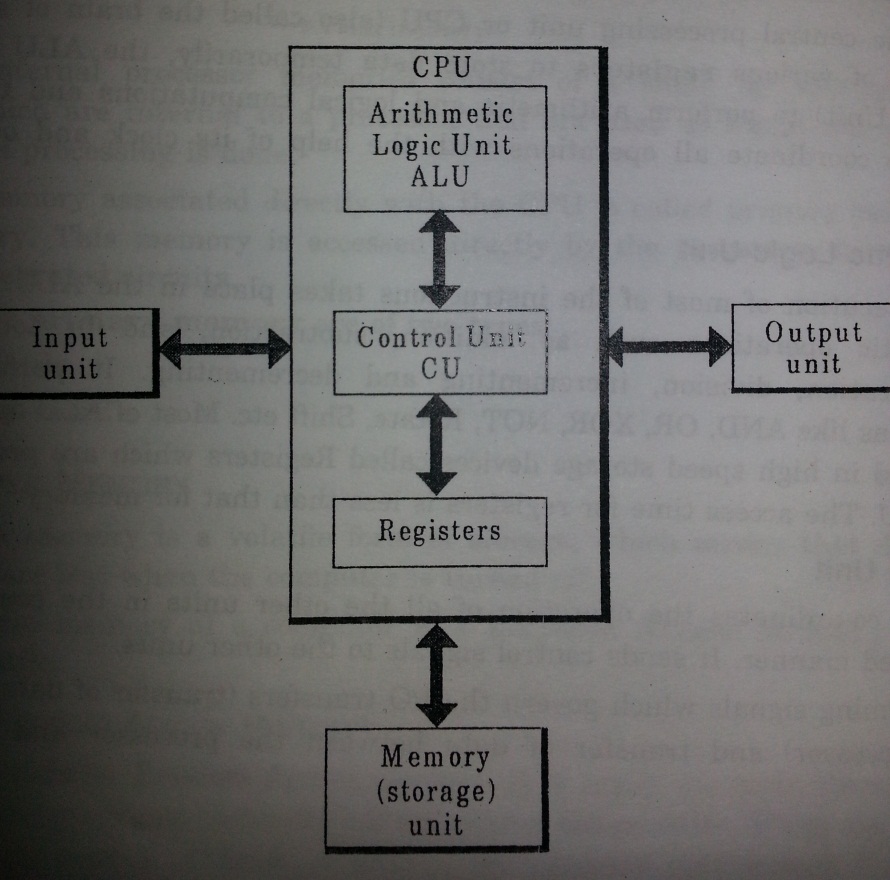
CPU which includes; Control unit (CU), Arithmetic and Logic unit (ALU), Registers.

Memory/Storage unit

Output unit

Buses





**NOTE: The main thing here is being able to list the cycle and explain each step**

**MAR**

* MAR is short for **memory address register**. This register has its output hooked up to the address bus. This register is going to be the only way for the CPU to communicate with the bus.
* The MAR can hold two different kinds of addresses. Either it stores the address of an instruction, or it stores the address of data. Of course, as far as memory is concerned, an address is an address. It doesn't care what is stored at that address.

**MDR**

* MDR is short for **memory data register**. As you might guess, the MDR is the analog of the MAR, except it is used with the data bus, instead of the address bus. Unlike the MAR, where we either place the address on the address bus or we don't (by deactivating the tri-state buffer between the MAR and the address bus), the MDR has more operations than just placing the data on the data bus or not placing it.
* In particular, data can go both ways. For a load operation, we expect to read data from the data bus into the MDR. For a store operation, we expect to place the data from the MDR onto the data bus.
* This means that the MDR can (parallel) load its data from one of two places: the data bus (for a load operation) and somewhere else in the CPU (for a store operation).

**NOTE: I don’t know why we need to know this, it is mentioned in the guide, I found this on google, feel free to go over it.**

**Input Unit**: Devices connected to the computer such as mouse & keyboard

Function:

* User feeds the data or programs through the input unit to the computer
* Converts data into corresponding code and sends it to the CPU for processing
* Converted data is also stored in memory unit

**Central Processing Unit (CPU):** Consists of various registers to store data temporarily:

* Arithmetic Logic Unit (ALU): completes logical and arithmetic computations
* Control Unit (CU): coordinates all operations with the help of its clock and other control circuits
* Communication bus system called the Primary memory bus and the address bus

**Arithmetic Logic Unit:** Where execution of most of the instructions takes place.

* Performs arithmetic operations like addition & subtraction etc.
* Performs logical operations like AND, OR, NOT etc.
* Most of the ALU operations can be stored in high speed storage devices called registers which are provided inside the CPU (the access time for registers is less than memory).

[**Accumulator**](http://en.wikipedia.org/wiki/Accumulator_%28computing%29)

* The Accumulator is the calculating unit of the ALU, it holds a temporary data within the ALU. It holds the ongoing total of the calculations being performed.

[**Instruction Register**](http://en.wikipedia.org/wiki/Instruction_register)

* The Instruction Register (IR) holds the actual instruction (opcode) which is being executed by the Processor. (it only holds the current instruction)
* The Opcode (Operation Code) is a part of the instruction that specifies the current operation.

[**Program Counter**](http://en.wikipedia.org/wiki/Program_counter)

* The Program Counter holds the Address of the next instruction. When the previous instruction is completed the instruction given in the PC linked instruction is executed.

**NOTE: Drawing the Block diagram and understanding what each component does is important!!!!**

**Control Unit (CU):** Coordinates the operation of all other units in the computer in an organized manner.

* Sends control signals to other units e.g. it generates control signals that which select ALU operations & CPU registers
* Moves data in and out of the CPU via the data bus
* Accesses memory addresses by using memory address bus

**Registers:** fast memory units that hold data & instructions during execution.

* They work at the speed of the CPU
* The size of the registers is called the word size(determines the amount of data that can processed per time e.g. 32bit & 64bit processors

**Memory unit:** where data is stored

**Stored program concept:** John von Neumann - program instructions can be coded and stored together in memory with coded data.

**Main/Primary memory:** stores programs and data during execution.

* RAM (random access memory) & ROM (read only memory)
* Contains a large number of semi-conductor storage cells each capable of storing 1 bit of information
* These cells are read or written in groups of fixed size called words.
* The number of bits is called the word length
* Each memory word is identified by a unique address
* Accessed directly by the processor

**Secondary memory:** is a permanent storage device

* Used to store large amounts of data permanently, especially if data is not accessed frequently

**Output Unit**: sends processed results to an output device.

**Computer memory:** the memory unit is that part of the computer which holds data and instructions for processing. The main functions are:

* To store both data & instructions given to the computer for processing
* To store the partial or complete results during processing

**Types of primary memory: (cache, virtual, RAM, ROM are all types of primary memory)**

* Primary memory is used by the computer to store currently executing programs andthe data used by these program. The program instructions and data are represented in memory as binary machine code by electronically indicating a 0 or a.
* Volatile memory: data and instructions are lost when the computer is turned off
* The main memory of a computer is a Random Access Memory, which is volatile

**RAM:** main storage & place where the programs and software are stored temporarily

* When the CPU runs a program, it fetches the program instructions from RAM and carries them out
* If the CPU needs to store the results of calculations, it can store them in RAM
* Read/write memory
* Fixed length words
* Access time = time required to locate and read a word from memory.
* The information in the memory can be located by a procedure called random access, where any location may be selected at random, therefore data can be accessed at high speeds
* CPU can read/write from the RAM, the speed is slow because of the CPU's speed is faster than the RAM's. This can be fixed by using Cache

**2 main types of RAM:**

**Static RAM (SRAM):** Some SRAM can be placed between main RAM and the processor as a temporary store for blocks of program instructions

**Dynamic RAM (DRAM)**

**Cache memory**: is RAM that a CPU can access more quickly than it can access regular RAM.

* As the CPU processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do the more time-consuming reading of data from larger memory.
* The cache on the microprocessor is referred to as level 1 cache and the cache between the main memory and processor is referred to as level 2.

**Virtual Memory**: For some programs the primary memory is not large enough to hold the whole program at the same time.

* Random access secondary memory (hard disc) can be used as an extension of the primary memory. This area is called the swap file.

**Non-Volatile memory (example:ROM):** retain their contents, even after the computer has switched off

* They store data permanently
* They can only be read from, not written into

**ROM**: the information stored can be permanent or semi-permanent.

**Long data life**

**Non volatile**

**PROM (Programmable ROM)**

**EPROM (Erasable PROM)**

**EEPROM (Electronically Erasable PROM**

**NOTE: you don’t need to know much about RAM & ROM, just know what they are and the differences. Learning the rest is up to you**

**Secondary memory:** refers to permanent memory storage such as disks and tape.

* + Data stored in primary memory must be written to secondary memory if it is to be saved permanently
  + Backing storage is required to keep in which is not needed in memory all of the time and which may be too large to fit into the memory of the computer (both programs and data are held on backing storage

**Two main ways of storing data on backing storage:**

**Sequential Access**

* + On for example, Tapes information is stored in a line (one bit of information after another). To find the information you need to read one bit after another to locate the information.

**Direct Access**

* + Means that data can be located anywhere on the disk without having to shuffle through all the other data. This is accomplished differently depending on the form of storage.

**Bits & Bytes:** Computers are binary devices.

They operate by storing instructions and data as sequences of 1's and 0's.

Computers work with a fixed number of bits usually multiples of 8-bits, this is know as a word

**Machine Instruction Cycle:** When a computer program is stored as a series of instructions in machine code in primary memory the following steps are carried out:

